Model validation

Anders Nielsen & Ben Bolker

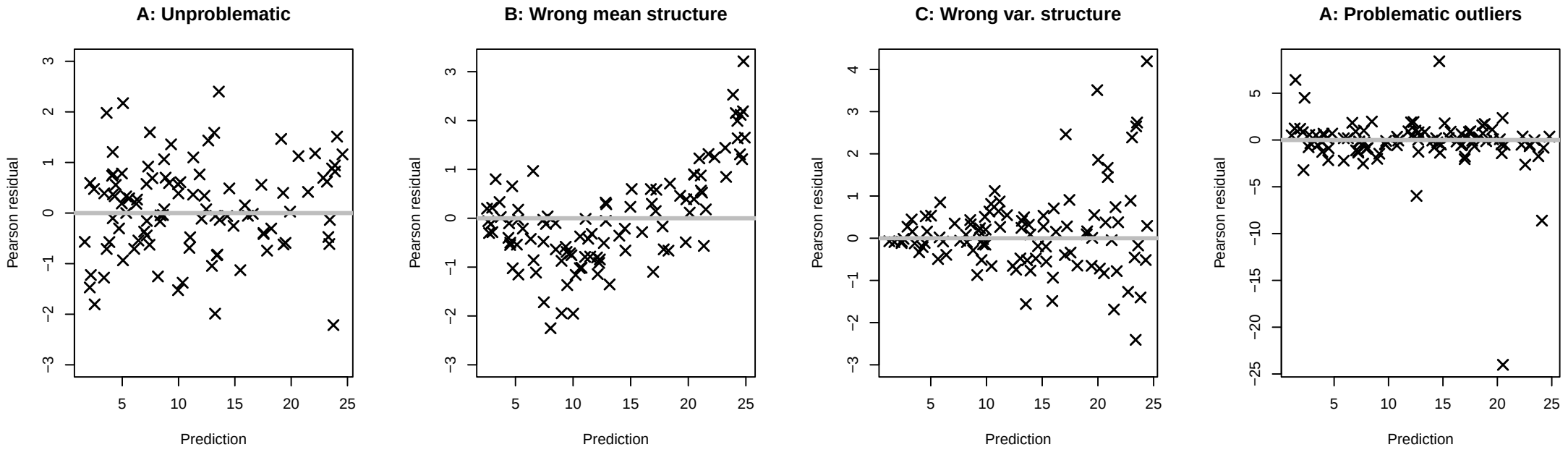
an@aqua.dtu.dk

Residuals (Pearson)

- Residuals are classically defined as:

$$r_i = \frac{\text{obs}_i - \text{pred}_i}{\text{sd}(\text{obs}_i)}$$

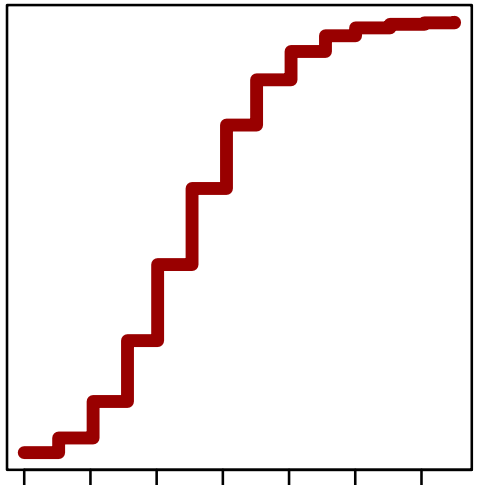
- What are we looking for in residuals?



Residuals

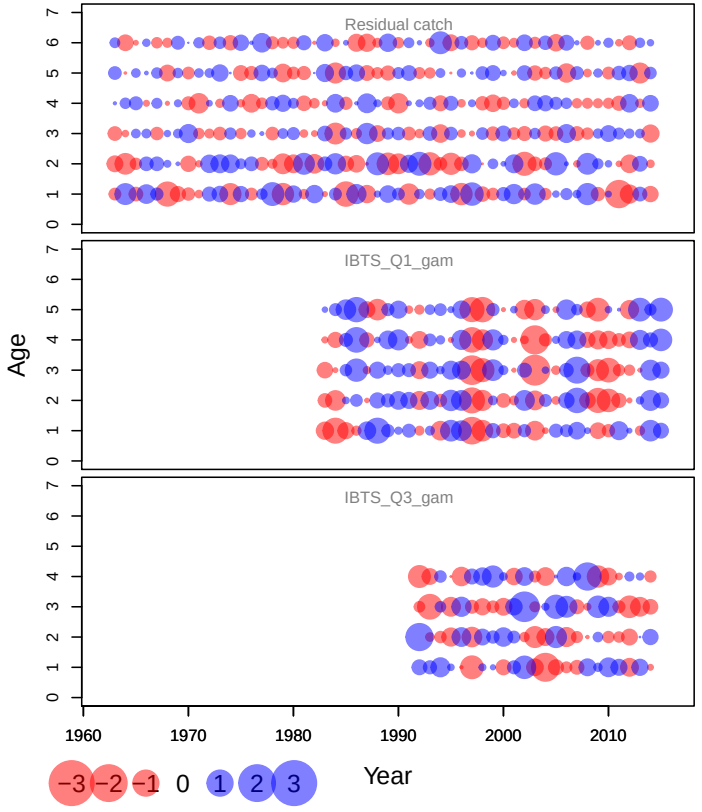
- In state-space models, and other models with correlated observations, residuals calculated as $r_i = (y_i - \hat{y}_i)/\hat{\sigma}_i$ are not supposed to be independent $N(0,1)$ even in perfectly correct models.
- A safer alternative is the **one-observation-ahead** residuals $(y_i - \hat{y}_{i|i-1})/\hat{\sigma}_{i|i-1}$.
- More generally the **one-observation-ahead-quantile-residuals**

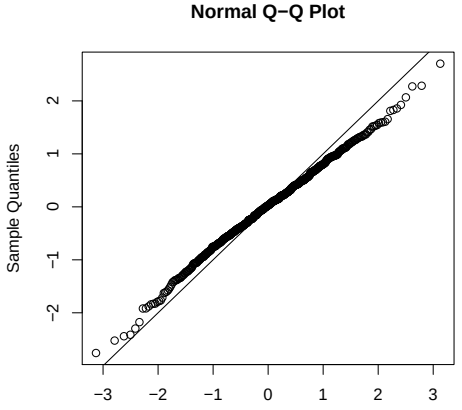
$$\Phi^{-1}(P(Y_i <= y_i | Y_{i-1} = y_{i-1} \ldots Y_1 = y_1))$$



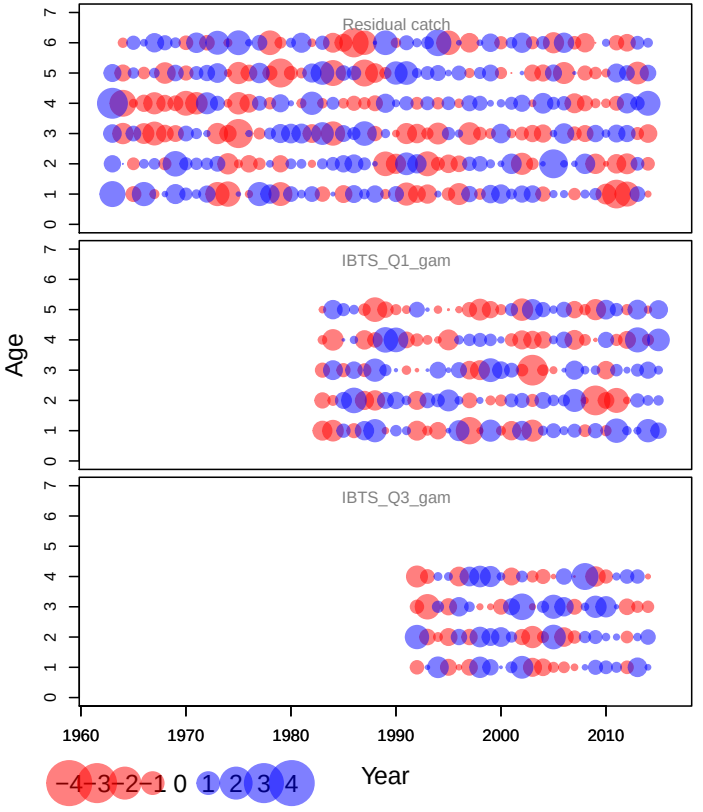
- Randomized if originating from a discrete distribution
- Requires extra work when the model is solved via Laplace approximation
- But it does matter — the residuals are different.

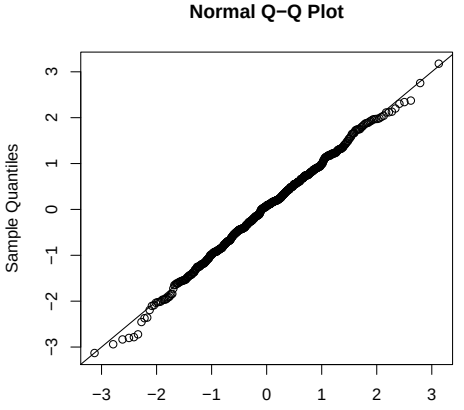
Wrong $(y_i - \hat{y}_i)/\hat{\sigma}_i$





Right $\Phi^{-1}(P(Y_i <= y_i | Y_{i-1} = y_{i-1} \ldots Y_1 = y_1))$





Exercise (skip): Same in independent Gaussian model?

- Compare the Pearson with the one-step-ahead residuals in the basic assessment model (`fsa.R`)
- First optimize the model, and remember to 'flag' the observation vector with

```
logObs <- OBS(logObs)
```

- Then the one-step-ahead residuals can be computed with:

```
res <- oneStepPredict(obj)
```

For discrete observations (randomization)

- Let $x_i \sim$ pois(λ) (with a c.d.f. P)
- Define $u_i \sim$ unif$(P(x_i - 1), P(x_i))$
- Define $z_i = \Phi^{-1}(u_i)$
- Now $z_i \sim \mathcal{N}(0, 1)$

```
# observations
x <- rpois(1000,3)
ppois.u <- function(x, lambda){
  runif(length(x), ppois(x-1,lambda), ppois(x,lambda)) #uses the fact that ppois(-1,lambda)=0
}
U <- ppois.u(x,3)
Z <- qnorm(U)
```

rand.R

Mini exercise: Repeat this example with a different distribution (e.g. Negative binomial) to see that you can in fact get perfect $N(0, 1)$ residuals using this approach.

Exercise (joint): AR1-Poisson example

On a scientific survey trip the number of Mackerel eggs were counted per haul along a path. In total 100 equidistant positions were sampled and the observations $y_1, y_2, \ldots, y_{100}$ were recorded. The counts are assumed to be related to the density at the positions and hence autocorrelated along the path. The model used to describe the observations is the state space model:

$$y_i \sim \mathcal{P}(e^{\gamma_i}), \text{ where}$$
$$(\gamma_i - \mu) \sim \mathcal{N}(\phi(\gamma_{i-1} - \mu), \sigma^2)$$

where γ is the random AR(1) process representing the true intensity of mackerel eggs. The model can be implemented and the parameters estimated as in the file `ar1.R`

- Implement the model and compute the one-obs-ahead-residuals
- Try to change the observations to violate the model assumptions

Exercise: Residuals in a GLMM

- With data from Elston et al. (2001) we look at counts of ticks y_i, $i = 1, \ldots, n$. A model suggested is:

$$
\begin{aligned}
y_i &\sim \mathcal{P}(\lambda_i), \text{ where} \\
\log(\lambda_i) &= \alpha \cdot (\texttt{HEIGHT}_i - \overline{\texttt{HEIGHT}}) + \beta(\texttt{YEAR}_i) + L(\texttt{LOCATION}_i) + B(\texttt{BROOD}_i) + I(\texttt{INDEX}_i), \text{ where} \\
L_l &\sim \mathcal{N}(0, \sigma_L^2),\ B_b \sim \mathcal{N}(0, \sigma_B^2), \text{ and } I_i \sim \mathcal{N}(0, \sigma_I^2) \text{ are all independent.}
\end{aligned}
$$

- On the next page is a pedestrian implementation of the model. Compute one-observation-ahead residuals for this model.
- Plot them with the log-prediction on the x-axis
- Could we do the same with non-standard models ...?

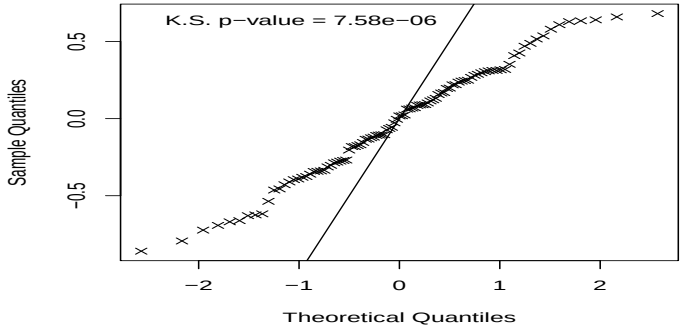
```r
library(RTMB)

tick <- read.table("Elston2001_tickdata.txt", header=TRUE, colClasses=c("factor","numeric","factor","numeric","factor","factor"))
tick$cHEIGHT <- tick$HEIGHT-mean(tick$HEIGHT)

par <- list(alpha=0, beta=numeric(nlevels(tick$YEAR)), logSdI=0, logSdB=0, logSdL=0,
            zI=numeric(nlevels(tick$INDEX)), zB=numeric(nlevels(tick$BROOD)), zL=numeric(nlevels(tick$LOCATION)))

f <- function(par){
  getAll(par,tick)
  jnll <- 0
  jnll <- jnll - sum(dnorm(zI, 0, sd=exp(logSdI),log=TRUE))
  jnll <- jnll - sum(dnorm(zB, 0, sd=exp(logSdB),log=TRUE))
  jnll <- jnll - sum(dnorm(zL, 0, sd=exp(logSdL),log=TRUE))
  logLambda <- alpha*cHEIGHT+beta[YEAR]+zI[INDEX]+zB[BROOD]+zL[LOCATION]
  jnll <- jnll - sum(dpois(TICKS, exp(logLambda), log=TRUE))
  jnll
}

obj <- MakeADFun(f, par, random=c("zI","zB","zL"), silent=TRUE)
fit <- nlminb(obj$par, obj$fn, obj$gr)

# same as:
# glmmTMB::glmmTMB(TICKS ~ -1+as.factor(YEAR)+cHEIGHT+(1|LOCATION)+(1|BROOD)+(1|INDEX), family=poisson, data=tick)
```

files/ticks.R

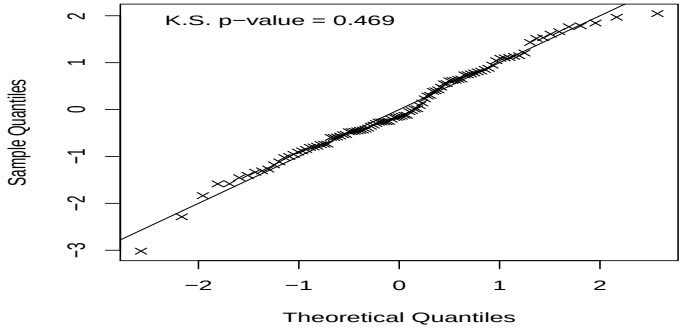
Process residuals

- Cannot just use the predicted random effects U, but
- If (Y, U) is distributed according to joint pdf. $L(y, u)$
- Observed y is then a sample from marginal distribution with pdf. $\int L(y, u)du$
- Generate one sample $u^\star$ from conditional distribution of $U|Y = y$
- Then the set $(y, u^\star)$ is a sample from joint distribution of (Y, U)
- Assumed distribution of $u^\star$ can be validated by standard tests

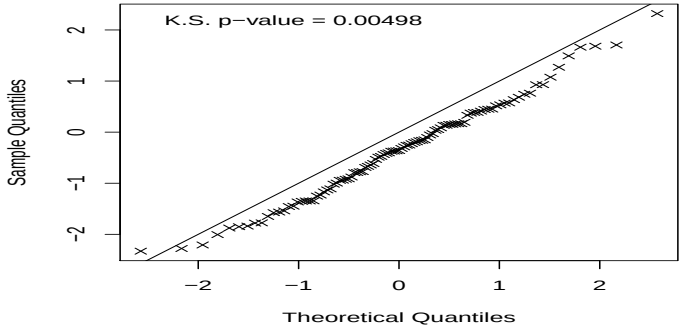
Wrong (using est. RE)



Right (joint sample)



Right. Model wrong



Code for the joint sample approach

```
sdr <- sdreport(obj)
estX <- summary(sdr,"random")
C <- solve(obj$env$spHess(obj$env$last.par.best, random=TRUE))
Xr <- MASS::mvrnorm(1,estX[,1],C)
```

Exercise: Calculate process residuals for the AR1-Poisson example. What distribution should we expect? Can we calculate quantities that we should expect are independent N(0,1)?

Check Laplace via simulation

- RTMB offers a very neat approach
- The expectation of the gradient of the negative log-likelihood is 0.

$$E_\theta \nabla \ell(\theta; X) = 0$$

- This means if we simulate from the model, then the average gradient should be zero.
- But this only holds for the real likelihood.
- So if the approximation is wrong, then the average gradient will not be zero
- We can simulate as many data sets as we wish, so we can test this.
- Notice: that even the smallest bias will be detected if we simulate enough
- Notice: Models with a modest bias can still be useful

```
$joint$p.value
[1] 0.4690289
...
$marginal$p.value
[1] 0.7745296
$marginal$bias
...
```

Appendix: The math for the Laplace checker

$$
\begin{aligned}
E_\theta\left(\nabla \ell(\theta; X)\right) &= \int P_\theta(x) \nabla \ell(\theta; x) dx \\
&= -\int P_\theta(x) \frac{1}{P_\theta(x)} \nabla P_\theta(X) dx \\
&= -\nabla \int P_\theta(x) dx \\
&= 0
\end{aligned}
$$